Interacted Multiple Ant Colonies Optimization Framework: an Experimental Study of the Evaluation and the Exploration Techniques to Control the Search Stagnation

Alaa Aljanaby [1], Ku Ruhana Ku-Mahamud [2], Norita Md. Norwawi [3]
[1]*College of Arts and Sciences, University of Nizwa, Oman*
[2]*College of Arts and Sciences, Universiti Utara Malaysia, Malaysia*
[3]*Faculty of Science and Technology, Universiti Sains Islam Malaysia, Malaysia*
alaa@unizwa.edu.om, ruhana@uum.edu.my, norita@usim.edu.my
doi: 10.4156/ijact.vol2.issue1.7

Abstract

Search stagnation is a serious problem that all Ant Colony Optimization (ACO) algorithms suffer from regardless of their application domain. The framework of Interacted Multiple Ant Colonies Optimization (IMACO) is a recent proposition. It divides the ants' population into several colonies and employs certain techniques to organize the work of these colonies. This paper proposes new effective evaluation and exploration techniques for IMACO and experimentally tests the stagnation behavior of IMACO. The performance of IMACO was demonstrated by comparing it with the best performing ant algorithms like Ant Colony System (ACS) and Max-Min Ant System (MMAS). The Computational results show the superiority of IMACO. The results comparison shows that IMACO with the proposed techniques suffers less from stagnation than the best known ant algorithms of ACS and MMAS.

Keywords: *Ant colony optimization, combinatorial optimization problems, search stagnation, pheromone evaluation, exploration, exploitation*

1. Introduction

One of the successful applications of swarm intelligence is the application of Ant Colony Optimization (ACO). Swarm intelligence is a field of artificial intelligence that studies the intelligent behavior of groups such as the behavior of natural systems of social insects like ants, bees, wasps, and termites. ACO is inspired from the foraging behavior of real ants. Using a combination of priori information (heuristics) about the candidate solutions quality of and posteriori information (pheromone) about the goodness of the previously obtained solutions is the key element of ACO success [1].

Combinatorial optimization problems are complex problems arise when the task is to find the best solution out of a huge number of existing solutions [2]. These problems have been successfully tackled by ACO. Traveling salesman problem (TSP), quadratic assignment problem, vehicle routing problem, job scheduling problem and network routing problem are some well known examples of these problems.

Several ant algorithms are presented in the literature among them Ant Colony System (ACS) and Max-Min Ant System (MMAS) the best performing algorithms [3, 4]. The performance of these algorithms is interesting. Nevertheless, these algorithms are still far from being perfect, these algorithms can get a good solution at the early stages of the search process but unfortunately all ants quickly converged to a single solution and then the algorithm is unable to improve that solution [5]. This is a common problem that all ACO algorithms suffer from regardless of the application domain; it is called search stagnation problem. The chance of stagnation proportionally increases with the increase of the problem size.

One new direction of ACO researches that focus on enhancing the performance of ACO and reducing the effect of the search stagnation is the use of Multiple Ant Colonies Optimization (MACO) where several ant colonies work together to collectively solve an optimization problem. MACO offers good opportunity to explore a large area of the search space and find (near-) optimal solution. MACO seems to be appropriate approach to improve the performance of ACO algorithms [6, 7]. IMACO follows this approach and tries to improve the performance of ACO algorithms by utilizing several ant colonies with certain techniques to organize the work of these colonies.

In this paper, the framework of IMACO is described in section 2. New pheromone evaluation and exploration techniques for IMACO are proposed in section 3. In section 4, an experimental study is conducted to test the effect of the proposed techniques. The performance of IMACO is compared with the performance of ACS and MMAS and the stagnation behavior of IMACO is analyzed and compared with the behavior of ACS and MMAS. The final section concludes the paper.

2. Interacted Multiple Ant Colonies Optimization

IMACO framework is recently proposed in pervious work of the authors [8, 9]. In this framework there are two levels of interaction the first one is the colony level and the second one is the population level. The colony level interaction can be achieved through the pheromone depositing process within the same colony; the pheromone updating mechanism is responsible for the implementation of this kind of interaction. The population level interaction is achieved by evaluating the pheromones of different colonies using some evaluation function; the responsibility here is of the pheromone evaluating mechanism.

The work activities of a single colony in the proposed IMACO algorithm are based on ACS. Each colony has its own pheromone that is used as an interaction between the ants of the same colony. The interaction between ant colonies using pheromone can be organized in different terms. The IMACO algorithm is described as follows. M colonies of m ants each are working together to solve some combinatorial problem. The probabilistic decision of the ant k belongs to the colony v to move from node i to node j is defined as:

$$j = \begin{cases} \arg \max_{l \in N_i^k} \{ f(P_{il})\, H_{il}^{\beta} \} & if \;\; q \le q_0 \\ S & \text{otherwise} \end{cases} \qquad (1)$$

The random variable S is selected according to the following probabilistic rule:

$$S = \begin{cases} \dfrac{f(P_{ij})\, H_{ij}^{\;\beta}}{\sum_{l \in N_i^{kv}} f(P_{il})\, H_{il}^{\;\beta}} & \text{if } j \in N_i^{kv} \\ 0 & \text{otherwise} \end{cases} \qquad (2)$$

Where N_i^{kv} is the set of remaining nodes to be visited by the k^{th} ant of colony v located at node i, P^v_{ij} is the pheromone of colony v on the edge $(i\ ,j)$ and $f(P_{ij})$ is the evaluation function of the pheromone on the edge (i, j) that will be discussed in Section 3.

Global and local pheromone updating are used in IMACO. Global pheromone updating includes that best ant of each colony deposits an amount of pheromone on its own path. The best ant refers to the ant that got the so far best (global) solution since the starting of the algorithm execution or the ant that got the best solution in the current iteration of the algorithm execution. In this work a combination of so far best and iteration best ants are allowed to update the pheromone.

After all ants of all colonies complete their tours (i.e., one algorithm iteration), the ant that finds the so far best solution in its colony is allowed to deposit an amount of the colony's pheromone on the edges of its tour according to the following global pheromone update:

$$P_{ij}^{v} = (1-\sigma) P_{ij}^{v} + \sigma \Delta p_{ij}^{v.bs} \qquad (3)$$

Where σ is a pheromone evaporation parameter its value is in the range [0, 1] and $\Delta P_{ij}^{v.bs}$ is the pheromone quantity added to the connection (i, j) belonging to the best solution of the v^{th} colony $L^{v.bs}$ and is given by:

$$\Delta p_{ij}^{v.bs} = \begin{cases} 1/L^{v.bs} & \text{if } (i,j) \text{ belongs to the best tour of colony } v \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

To create a search diversification IMACO uses iteration best solution once in the pheromone updating after each 50 times of using the global best solution [8, 9]. Local pheromone updating includes that each ants reduces the amount of pheromone on paths it uses in order to give a more chance to other paths to be chosen by the future generations. Local pheromone update is applied by each ant on the visited edges. It is very important rule as it is performed during the solution construction this helps to yield different pheromone evaluation values for the same edge in the same iteration at different solution construction steps and it is given by:

$$P_{ij}^{v} = (1-\gamma)P_{ij}^{v} + \gamma p_0 \tag{5}$$

Where P_0 is the initial pheromone value and γ is another pheromone evaporation parameter with a value in the range [0, 1].

3. The proposed Techniques

3.1. Pheromone evaluation technique

The proposed technique evaluating the pheromone as an average of the pheromone values of all colonies on some edge. This means that an ant will make its decision to choose some edge based on the average of the available experiences of ants of all colonies that visited this edge in the past. This variant of IMACO is referred hereafter as IMACO-AVG.

Given that for each edge there are M pheromone values each belongs to a single colony. Average pheromone evaluation function evaluates the pheromone on any edge as an average of the available M values. The average pheromone evaluation function $f(P_{ij})$ on the edge (i, j) for IMACO-AVG will be defined as:

$$f(P_{ij}) = \frac{\sum_{v=1}^{M} P_{ij}^{v}}{M} \tag{6}$$

The above is pure average evaluation that depends 100% on the average evaluation function [8, 9, 10]. The following new rule is a more general which evaluates the pheromone as a composition between the pheromone values of the ant own colony and the value of the pheromone evaluation function based on some pheromone evaluation rate. Consider that the composition rate is 0.5; an ant will build 50% of its decision based on its own colony's experience and the other 50% based on the experiences of other colonies. This new variant will be called IMACO-AVG E λ where λ is the pheromone evaluation rate; its value is in the range [0, 1]. The pheromone evaluation function is then defined as:

$$f'(P_{ij}) = \lambda P_{ij}^{s} + (1-\lambda)\ f(P_{ij}) \tag{7}$$

Where P_{ij}^{s} is the pheromone belongs to colony s on edge (i, j). Note that IMACO-AVG E0 represents the pure average pheromone evaluation and IMACO-AVG E1 represents no interaction between utilized ant colonies. Next section experimentally tests different values within the range [0, 1] for λ to find out the value that leads to the best performance of IMACO-AVG.

3.2. Exploration technique

Each ant makes a probabilistic decision when it needs to move to a new node. The probabilistic decision is based on heuristic information (cost) and pheromone information. Pheromone represents information about previous experiences of the ant's own colony and of the other colonies. While heuristic represent a priori information about the goodness of a solution. Exploration and exploitation is controlled by the parameter q_0 whose value is in [0, 1]. It is usually used in ant's probabilistic decision as trade-off between exploitation (choosing the edge with the higher value of the multiplication of pheromone and heuristic values) and exploration (choosing the edge randomly according to some probability distribution). Setting q_0 to zero means that the algorithm uses a pure exploration while pure exploitation is reached by setting q_0 to one. However, the value used for q_0 in many research papers usually between 0.5 and 0.9 [3, 4]. Most of the work done using ACS in solving different problems was with q_0 =0.9 which gives the algorithm a high chance of exploitation without loosing the chance of exploration.

As finding an effective exploration technique is one objective of this research, this section considers the case where different ants' colonies have different values for the parameter q_0. The value 0.8 has been assigned to the centre colony whose number equal to int (no. of colonies / 2). This value is increased / decreased for the colonies after / before the centre colony by a changing factor called QCF. The experimental values of QCF were 0.025, 0.02, 0.015 and 0.01. This technique enables the utilized ant colonies to work with different levels of exploration. Some will prefer high exploration of new areas of search space while other colonies will prefer high exploitation search history. The effect of this technique will be experimentally tested in next section.

4. Result and Discussion

TSP is a well known combinatorial optimization problem. Given that a certain number of nodes available, it is the problem of finding the shortest closed path that visit each node exactly once. TSP is usually used as a test bed for all new ant algorithms. All TSP instance used in this paper are taken from TSPLIB [10]. IMACO-AVG has been first tested using lin318 TSP benchmark instance that has 318 nodes and its optimal solution is 42029. Several experiments were run using 1 to 10 colonies. The results are averaged over 10 trials with 10000 iterations per trial. The parameters setting are β=2, $\sigma = \gamma$ = 0.1 and q_0 = 0.9. The heuristic function used for TSP is the inverse of the distance, i.e., $H_{ij}=1/d_{ij}$.

Figure 1 shows the results of using ACS and IMACO-AVG on lin318 TSP instance respectively using 10 to 100 ants. It is obvious that increasing the number of utilized ants for both experiments result in the decline of the performance of ACS. This means that ACS can not benefit form the increase in the number of utilized ants, the algorithm always get trapped in local bad optima and can not improve the solution quality. The better results obtained when the number of ants is 20-30.

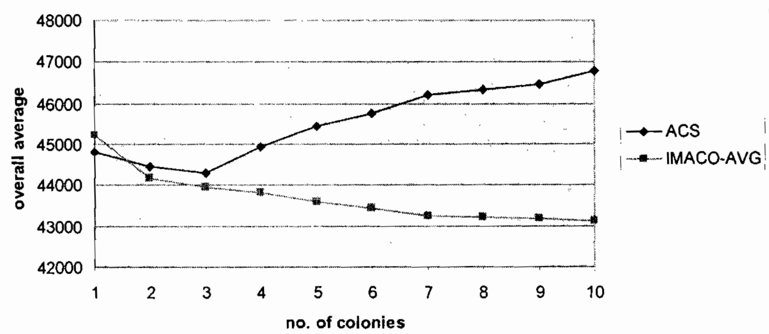

Figure 1. ACS and IMACO-AVG performance comparison

These results show that IMACO-AVG is able to improve its performance by increasing the number of utilized ants' colonies. The superior of IMACO-AVG is clear as this algorithm shows a stabilizing performance using the increased number of colonies. The average pheromone evaluation technique was a successful organizing technique of the ants activities up to 10 colonies utilized. However, given the stochastic nature of these algorithms it is better to set a range on the number of colonies that gave the best results which was 7-10 colonies.

To test the effect of the proposed pheromone evaluation technique, Figure 2 shows the results of several experiments ran using IMACO-AVG with 7-10 colonies using different values for the pheromone evaluation rate (λ=0, 0.1, 0.2, ..., 1). The best results obtained when λ=0.4. In this case 40% of the pheromone evaluation resulted value depends on the pheromone (experience) of ant's own colony and the 60% depends on the pheromone of all other colonies. The results were better than those obtained using the pure pheromone evaluation with λ=0.

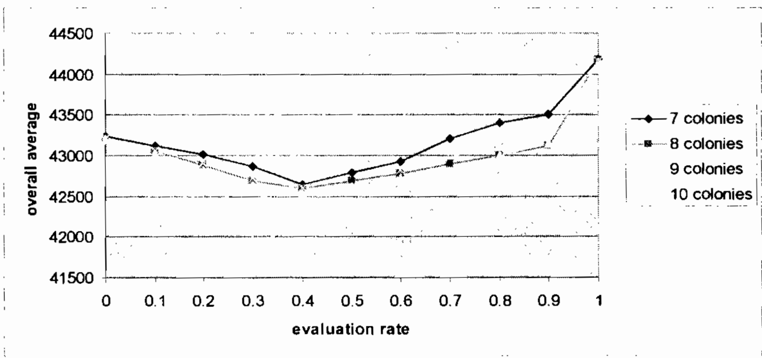

Figure 2. IMACO-AVG with 7-10 colonies using different pheromone evaluation rates

Fig. 3 shows the result of using IMACO-AVG with 7-10 colonies with different QCF values. The result with QCF=0 represent the original results obtained from previous experiment. IMACO-AVG results improved using this technique especially for QCF values 0.015, 0.02 and 0.025. However, the best result is obtained when QCF=0.025.

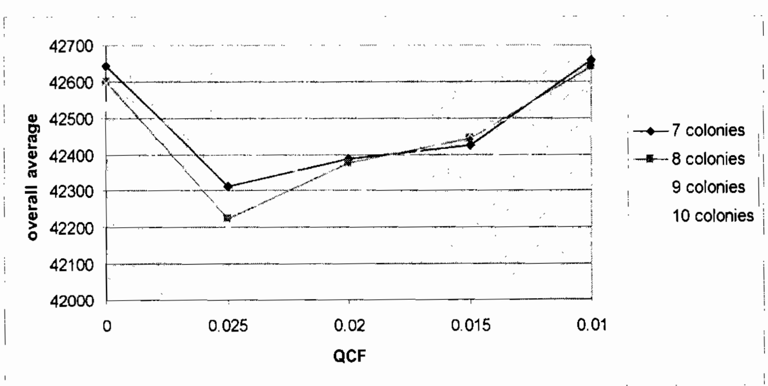

Figure 3. IMACO-AVG E0.4 7-10 colonies with different QCF values

4.1. Search stagnation analysis

The stagnation behavior of IAMCO is the main concern of this section. It is obvious that the results obtained in pervious sections support that idea that IMACO suffers less from stagnation than ACS through its ability to obtain better results than ACS. However, this section tries to experimentally prove this idea by observing 100 typical runs of ACS, MMAS, and IMACO-AVG and calculate how many times that each algorithm get trapped into stagnation situation.

As an initial step Table 1 shows the results of tracing one typical run of the four algorithms on lin318. ACS ran with 10 ants, MMAS with 318 ants, and IMACO-AVG E.4 with 9 colonies of 10 ants each. The results in Table 1 shows the so far best solution obtained after completing certain number of iterations.

Table 1. A One trial typical run on lin318

Iterations completed	ACS	MMAS	IMACO-AVG
10%	46145	45700	45423
20%	45832	45313	45001
30%	45312	44890	44530
40%	44933	44621	44050
50%	44565	44121	43765
60%	44014	43776	43278
70%	43614	43411	43090
80%	43614	43111	42813
90%	43614	42871	42719
100%	43614	42871	42590

The results of Table 1 are diagrammatically presented in Figure 4. It obvious that ACS can not improve its solution after 70% of algorithm iterations completed, the last 30% of the algorithm iterations ACS search process was stagnant. MMAS incorporated techniques to control stagnation did better than ACS but not better than IMACO-AVG. IMACO-AVG was the best among the other algorithms. It was able to direct its search away from stagnation situation and improve its solution.

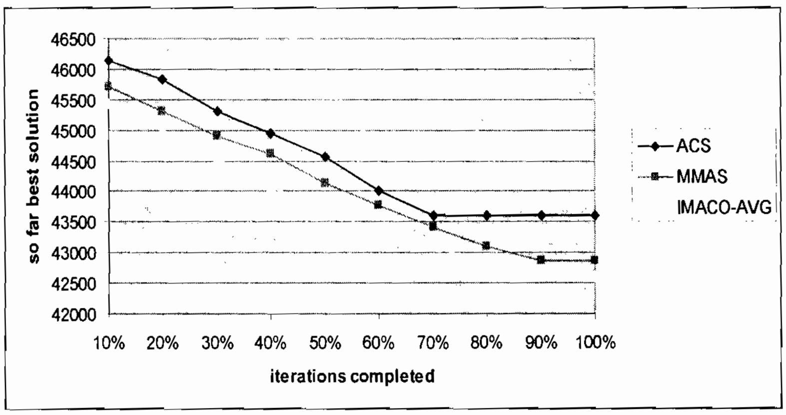

Figure 4. A One trial typical run on lin318

Table 2 shows a result of 100 of typical runs like those described above. The numbers in the table shows how many times the algorithm cannot improve its solution from the previous step. The algorithm which suffers more from stagnation was ACS. The chance of stagnation increases after 50% of iterations completed because all ants converged to one solution or one dominant path with highest amount of pheromone and ability to get out from this situation decrease with the time. IMACO-AVG was the algorithm that suffers less from stagnation. This proves that the techniques incorporated in this algorithm did well to avoid such situations.

Table 3 shows a comparison between the performance of ACS, MMAS and IMACO-AVG E.4 on five TSP instances. The results in this table are the best overall average. The name of each TSP instance is followed by the number of nodes in this instance and the optimal solution is given below the instance name. IMACO-AVG utilized 9 colonies and λ=0.4. The results of ACS and MMAS are taken from literature [11]. The results of Table 3 show that IMACO-AVG E .4 outperforms the ACS and MMAS the best performing ACO algorithms.

Table 2. Stagnation situations in 100 trials typical run on lin318

Iterations completed	ACS	MMAS	IMACO-AVG
10%	0	0	0
20%	0	0	0
30%	0	0	0
40%	2	0	0
50%	15	3	0
60%	30	7	3
70%	35	9	0
80%	45	17	4
90%	45	19	1
100%	45	19	3

Table 3. Best overall average of ACS, MMAS and IMACO-AVG

TSP instance	Algorithm	Best overall average
kroA100 Opt: 21282	ACS	21420.0
	MMAS	21291.6
	IMACO-AVG-E.4	21290.6
lin318 Opt: 42029	ACS	43296.85
	MMAS	42346.60
	IMACO-AVG-E.4	42191.6
pcb442 Opt: 50778	ACS	51935.6
	MMAS	51515.2
	IMACO-AVG-E.4	51100.0
att532 Opt: 27686	ACS	28522.8
	MMAS	28112.6
	IMACO-AVG-E.4	28018.3
ftv170 opt: 2755	ACS	2826.5
	MMAS	2817.7
	IMACO-AVG-E.4	2791.8

It is important to mention that all algorithms run exactly the same number of computation steps. The number of iterations each algorithm runs on each instance is equal to number of computation steps / number of ants. For example ACS with 10 ants on lin318 runs 318000 iterations per trial while IMACO-AVG with 9 colonies each of 10 ants on the same instance runs 35333 iterations per trial. Both algorithms run 3180000 computation steps by doing different number of iterations using different number of ants.

5. Conclusion

Pheromone evaluation technique is an essential component of IMACO framework. A more effective pheromone evaluation mechanism has been proposed. The proposed mechanism is a composition between the pheromone of ant's own colony and the average of the pheromone of all other colonies. A new exploration technique has also been proposed for IMACO. This technique lets different colonies work with different levels of exploration of the search space. IMACO with the new evaluation and exploration techniques has been tested on different TSP instances and results were compared with ACS and MMAS results and its superior performance was obvious. The experimental results show that IMACO was the algorithm that suffers less from stagnation than other well known ACO algorithms.

6. References

[1] C. Blum and A. Roli, "Meta-heuristics in combinatorial optimization: Overview and conceptual comparisons", ACM Computing Surveys, Vol. 35, No.3, 2003, pp. 268-308.
[2] M. Dorigo, M. Birattari and T. Stützle, "Ant colony optimization", IEEE Computational Intelligence Magazine, Nov. 2006.
[3] M. Dorigo, and T. Stützle, "The Ant Colony Optimization Meta-heuristic: Algorithms, Applications, and Advances", In: Handbook of Meta-heuristics (Eds. F. Glover and G. Kochenberger), 2002, pp. 250-285, Kluwer Academic Publishers.
[4] M. Dorigo, and T. Stützle, "Ant colony optimization", London: The MIT Press.
[5] C. Blum, and M. Dorigo, "Search bias in ant colony optimization: On the role of competition-balanced systems", IEEE Trans. on Evolutionary Computation, Vol. 9, No. 2, 2005, pp. 159-174.
[6] H. Kawamura, M. Yamamoto, K. Suzuki, and A. Ohuchi, "Multiple ant colonies algorithm based on colony level interactions", IEICE Trans. Fundamentals, Vol. E83-A, No.2, 2000, pp. 371-379.
[7] J. Jong, and M. Wiering, "Multiple ant colony system for the bus-stop allocation problem", Proc. of the Thirteenth Belgium-Netherlands Conference on Artificial Intelligence (BNAIC'01), Amsterdam, Netherlands, 2001, pp. 141–148.
[8] A. Aljanaby, K.R. Ku-Mahamud, and N.M. Norwawi, "Optimizing large scale problems using multiple ant colonies algorithm based on pheromone evaluation technique", International Journal of Computer Science and Network Security (IJCSNS), Vol. 8, No, 10, 2008, pp. 54-58.
[9] A. Aljanaby, K.R. Ku-Mahamud, and N.M. Norwawi, "A new multiple ant colonies optimization algorithm utilizing average pheromone evaluation mechanism", Proc. of the Knowledge Management International Conference (KMICe'08), Langkawi, Malaysia, 2008, pp. 531-534.
[10] TSPLIB, http://www.iwr.uni-heidelberg.de/groups /comopt/software/TSPLIB95/tsp/.
[11] T. Stützle, and H. Hoos, "Max-Min Ant System", Journal of Future Generation Computer Systems, Vol. 16, No. 8, 2000, pp. 889–914.